

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

# **Improved Stealthy Audio Watermarking**

Inventors:

Darko Kirovski  
Henrique Malvar

ATTORNEY'S DOCKET NO. MS1-570US

1 **CROSS-REFERENCE TO RELATED APPLICATIONS**

2 This application claims priority from U.S. Provisional Patent Application  
3 Serial No. 60/143432 entitled "Improved Audio Watermarking" filed on July 13,  
4 1999.  
5

6 **TECHNICAL FIELD**

7 This invention relates to protecting audio content by using watermarks.  
8 More particularly, this invention relates to a set of techniques that improve  
9 encoding and decoding of audio watermarks.  
10

11 **BACKGROUND**

12 Since the earliest days of human civilization, music has existed at the  
13 crossroads of creativity and technology. The urge to organize sound has been a  
14 constant part of human nature, while the tools to make and capture the resulting  
15 music have evolved in parallel with human mastery of science.  
16

17 Throughout the history of audio recordings, the ability to store and transmit  
18 audio (such as music) has quickly evolved since the early days just 130 years ago.  
19 From Edison's foil cylinders to contemporary technologies (such as DVD-Audio,  
20 MP3, and the Internet), the constant evolution of prerecorded audio delivery has  
21 presented both opportunity and challenge.

22 Music is the world's universal form of communication, touching every  
23 person of every culture on the globe. Behind the music is a growing multi-billion  
24 dollar per year industry. This industry, however, is constantly plagued by lost  
25 revenues due to music piracy.

1 **Protecting Rights**

2 Piracy is not a new problem. However, as technologies change and  
3 improve, there are new challenges to protecting music content from illicit copying  
4 and theft. For instance, more producers are beginning to use the Internet to  
5 distribute music content. In this form of distribution, the content merely exists as  
6 a bit stream which, if left unprotected, can be easily copied and reproduced.

7  
8 At the end of 1997, the International Federation of the Phonographic  
9 Industry (IFPI), the British Phonographic Industry, and the Recording Industry  
10 Association of America (RIAA) engaged in a project to survey the extent of  
11 unauthorized use of music on the Internet. The initial search indicated that at any  
12 one time there could be up to 80,000 infringing MP3 files on the Internet. The  
13 actual number of servers on the Internet hosting infringing files was estimated to  
14 2,000 with locations in over 30 countries around the world.

15 Each day, the wall impeding the reproduction and distribution of infringing  
16 digital audio clips (e.g., music files) gets shorter and weaker. "Napster" is an  
17 example of an application that is weakening the wall of protection. It gives  
18 individuals access to one another's MP3 files by creating a unique file-sharing  
19 system via the Internet. Thus, it encourages illegal distribution of copies of  
20 copyrighted material.

21 As a result, these modern digital pirates effectively rob artists and authors  
22 of music recordings of their lawful compensation. Unless technology provides for  
23 those who create music to be compensated for it, both the creative community and  
24 the musical culture at large will be impoverished.  
25

## **Identifying a Copyrighted Work**

Unlike tape cassettes and CDs, a digital music file has no jewel case, label, sticker, or the like on which to place the copyright notification and the identification of the author. A digital music file is a set of binary data without a detectible and unmodifiable label.

Thus, musical artists and authors are unable to inform the public that a work is protected by adhering a copyright notice to the digital music file. Furthermore, such artists and authors are unable to inform the public of any addition information, such as the identity of the copyright holder or terms of a limited license.

## **Digital Tags**

The music industry and trade groups were especially concerned by digital recording because there is no generation loss in digital transfers—a copy sounds the same as the original. Without limits on unauthorized copying, a digital audio recording format could easily encourage the pirating of master-quality recordings.

One solution is to amend an associated digital “tag” with each audio file that identified the copyright holder. To implement such a plan, all devices capable of such digital reproduction must faithfully reproduce the amended, associated tag.

With the passage of the Audio Home Recording Act of 1992, inclusion of serial copying technology became law in the United States. This legislation mandated the inclusion of serial copying technology, such as SCMS (Serial Copy Management System), in consumer digital recorders. SCMS recognizes a “copyright flag” encoded on a prerecorded original (such as a CD), and writes that

1 flag into the subcode of digital copies (such as a transfer from a CD to a DAT  
2 tape). The presence of the flag prevents an SCMS-equipped recorder from  
3 digitally copying the copy, thus breaking the chain of perfect digital cloning.

4 However, subsequent developments—both technical and legal—have  
5 demonstrated the limited benefits of this legislation. While digital secure music  
6 delivery systems (such as SCMS) are designed to support the rights of content  
7 owners in the digital domain, the problem of analog copying requires a different  
8 approach. In the digital domain, information about the copy status of a given piece  
9 of music may be carried in the subcode, which is separate information that travels  
10 along with the audio data. In the analog domain, there is no subcode; the only  
11 place to put the extra information is to hide it within the audio signal itself.

### 12 13 Digital Watermarks

14 Techniques for identifying copyright information of digital audio content  
15 that address both analog and digital copying instances have received a great deal  
16 of attention in both the industrial community and the academic environment. One  
17 of the most promising “digital labeling” techniques is augmentation of a digital  
18 watermark into the audio signal itself by altering the signal’s frequency spectrum  
19 such that the perceptual characteristics of the original recording are preserved.

20 In general, a “digital watermark” is a pattern of bits inserted into a digital  
21 image, audio, or video file that identifies the file's copyright information (author,  
22 rights, etc.). The name comes from the faintly visible watermarks imprinted on  
23 stationery that identify the manufacturer of the stationery. The purpose of digital  
24  
25

1 watermarks is to provide copyright protection for intellectual property that is in  
2 digital format.

3 Unlike printed watermarks, which are intended to be somewhat visible,  
4 digital watermarks are designed to be completely invisible, or in the case of audio  
5 clips, inaudible. Moreover, the actual bits representing the watermark must be  
6 scattered throughout the file in such a way that they cannot be identified and  
7 manipulated. And finally, the digital watermark must be robust enough so that it  
8 can withstand normal changes to the file, such as reductions from lossy  
9 compression algorithms.

10 Satisfying all these requirements is no easy feat, but there are several  
11 competing technologies. All of them work by making the watermark appear as  
12 noise—that is, random data that exists in most digital files anyway. To view a  
13 watermark, you need a special program or device (i.e., a “detector”) that knows  
14 how to extract the watermark data.

15 Herein, such a digital watermark may be simply called a “watermark.”  
16 Generically, it may be called an “information pattern of discrete values.” The  
17 audio signal (or clip) in which a watermark is encoded is effectively “noise” in  
18 relation to the watermark.

## 19 20 **Watermarking**

21 Watermarking gives content owners a way to self-identify each track of  
22 music, thus providing proof of ownership and a way to track public performances  
23 of music for purposes of royalty distribution. It may also convey instructions,  
24 which can be used by a recording or playback device, to determine whether and  
25

1 how the music may be distributed. Because that data can be read even after the  
2 music has been converted from digital to an analog signal, watermarking can be a  
3 powerful tool to defeat analog circumvention of copy protection.

4 The general concept of watermarking has been around for at least 30 years.  
5 It was used by companies (such as Muzak™) to *audibly* identify music delivered  
6 through their systems. Today, however, the emphasis in watermarking is on  
7 *inaudible* approaches. By varying signals embedded in analog audio programs, it  
8 is possible to create patterns that may be recognized by consumer electronics  
9 devices or audio circuitry in computers.

10 For general use in the record industry today, watermarking must be  
11 completely inaudible under all conditions. This guarantees the artistic integrity of  
12 the music. Moreover, it must be robust enough to survive all forms of attacks. To  
13 be effective, watermarks must endure processing, format conversion, and  
14 encode/detect cycles that today's music may encounter in a distribution  
15 environment that includes radio, the Web, music cassettes, and other non-linear  
16 media. In addition, it must endure malevolent attacks by digital pirates.

### 17 Watermark Encoding

18 Typically, existing techniques for encoding a watermark within discrete  
19 audio signals facilitate the insensitivity of the human auditory system (HAS) to  
20 certain audio phenomena. It has been demonstrated that, in the temporal domain,  
21 the HAS is insensitive to small signal level changes and peaks in the pre-echo and  
22 the decaying echo spectrum.  
23  
24  
25

1 The techniques developed to facilitate the first phenomenon are typically  
2 not resilient to de-synch attacks. Due to the difficulty of the echo cancellation  
3 problem, techniques that employ multiple decaying echoes to place a peak in the  
4 signal's cepstrum can hardly be attacked in real-time, but fairly easy using an off-  
5 line exhaustive search. (The term "cepstrum" is the accepted terminology for the  
6 inverse Fourier transform of the logarithm of the power spectrum of a signal.)

7 Watermarking techniques that embed secret data in the frequency domain  
8 of a signal facilitate the insensitivity of the HAS to small magnitude and phase  
9 changes. In both cases, a publisher's secret key is encoded as a pseudo-random  
10 sequence that is used to guide the modification of each magnitude or phase  
11 component of the frequency domain. The modifications are performed either  
12 directly or shaped according to the signal's envelope.

13 In addition, watermarking schemes have been developed which facilitate  
14 the advantages but also suffers from the disadvantages of hiding data in both the  
15 time and frequency domain. It has not been demonstrated whether spread-  
16 spectrum watermarking schemes would survive combinations of common attacks:  
17 de-synchronization in both the temporal and frequency domain and mosaic-like  
18 attacks.

## 19 20 **Watermark Detection**

21 The copy detection process is performed by synchronously correlating the  
22 suspected audio clip with the watermark of the content publisher. A common  
23 pitfall for all watermarking systems that facilitate this type of data hiding is  
24 intolerance to desynchronization attacks (e.g., sample cropping, insertion,  
25



1 repetition, variable pitch-scale and time-scale modifications, audio restoration, and  
2 arbitrary combinations of these attacks) and deficiency of adequate techniques to  
3 address this problem during the detection process.  
4

### 5 **Desiderata of Watermarking Technology**

6 Watermarking technology has several highly desirable goals (i.e.,  
7 desiderata) to facilitate protection of copyrights of audio content publishers.  
8 Below are listed several of such goals.

9 Perceptual Invisibility. The embedded information should not induce  
10 audible changes in the audio quality of the resulting watermarked signal. The test  
11 of perceptual invisibility is often called the "golden ears" test.

12 Statistical Invisibility. The embedded information should be quantitatively  
13 imperceptible for any exhaustive, heuristic, or probabilistic attempt to detect or  
14 remove the watermark. The complexity of successfully launching such attacks  
15 should be well beyond the computation power of publicly available computer  
16 systems.

17 Tamperproofness. An attempt to remove the watermark should damage the  
18 value of the music well above the hearing threshold.

19 Cost. The system should be inexpensive to license and implement on both  
20 programmable and application-specific platforms.

21 Non-disclosure of the Original. The watermarking and detection protocols  
22 should be such that the process of proving audio content copyright both in-situ and  
23 in-court, does not involve usage of the original recording.  
24  
25

1        Enforceability and Flexibility. The watermarking technique should provide  
2 strong and undeniable copyright proof. Similarly, it should enable a spectrum of  
3 protection levels, which correspond to variable audio presentation and  
4 compression standards.

5        Resilience to Common Attacks. Public availability of powerful digital  
6 sound editing tools imposes that the watermarking and detection process is  
7 resilient to attacks spawned from such consoles. The standard set of plausible  
8 attacks is itemized in the Request for Proposals (RFP) of IFPI (International  
9 Federation of the Phonographic Industry) and RIAA (Recording Industry  
10 Association of America). The RFP encapsulates the following security  
11 requirements:

- 12        • two successive D/A and A/D conversions,
- 13        • data reduction coding techniques such as MP3,
- 14        • adaptive transform coding (ATRAC),
- 15        • adaptive subband coding,
- 16        • Digital Audio Broadcasting (DAB),
- 17        • Dolby AC2 and AC3 systems,
- 18        • applying additive or multiplicative noise,
- 19        • applying a second Embedded Signal, using the same system, to a  
20 single program fragment,
- 21        • frequency response distortion corresponding to normal analogue  
22 frequency response controls such as bass, mid and treble controls,  
23 with maximum variation of 15 dB with respect to the original signal,  
24 and
- 25

- applying frequency notches with possible frequency hopping.

### **Watermark Circumvention**

If the encoding of a watermark can thwart a malicious attack, then it can avoid the harm of the introduction of unintentional noise. Therefore, any advancement in watermark technology that makes it more difficult for a malevolent attacker to assail the watermark also makes it more difficult for a watermark to be altered unintentionally.

In general, there are two common classes of malevolent attacks:

1. De-synchronization of watermark in digital audio signals. These attacks alter audio signals in such a way to make it difficult for the detector to identify the location of the encoded watermark codes.
2. Removing or altering the watermark. The attacker discovers the location of the watermark and intentionally alters the audio clip to remove or deteriorate a part of the watermark or its entirety.

### **Framework to Thwart Attacks**

Accordingly, there is a need for a new framework of protocols for hiding and detecting watermarks in digital audio signals that are effective against malevolent attacks. The framework should possess several attributes that further the desiderata of watermark technology, described above. For example, such desiderata include “perceptual invisibility” and “statistical invisibility”. The framework should be tamperproof and inexpensive to license and implement on both programmable and application-specific platforms. The framework should be

1 such that the process of proving audio content copyrights both in-situ and in-court  
2 does not involve usage of the original recording.

3 The framework should also be flexible to enable a spectrum of protection  
4 levels, which correspond to variable audio presentation and compression  
5 standards, and yet resilient to common attacks spawned by powerful digital sound  
6 editing tools.

7 In addition, the framework will facilitate search for the "El Dorado" and  
8 the "Holy Grail" of watermarking technology.

9 The seemingly unattainable "El Dorado" of watermarking technology is an  
10 encoded watermark that is unalterable, irremovable, and cannot be de-synced  
11 without perceptually and noticeably affecting the audio quality.

12 Likewise, the seemingly unattainable "Holy Grail" of watermarking  
13 technology is an encoded watermark where a malevolent attacker may know how  
14 the watermark is encoded, but still cannot effectively attack it without perceptually  
15 and noticeably affecting the audio quality.

## 16 17 **SUMMARY**

18 Described herein is an audio watermarking technology for inserting and  
19 detecting watermarks in audio signals, such as a music clip. The watermark  
20 identifies the content producer, providing a signature that is embedded in the audio  
21 signal and cannot be removed. The watermark is designed to survive all typical  
22 kinds of processing, including compression, equalization, D/A and A/D  
23 conversion, recording on analog tape, and so forth. It is also designed to survive  
24 malicious attacks that attempt to remove or modify the watermark from the signal,  
25

1 including changes in time and frequency scales, pitch shifting, and cut/paste  
2 editing.

3 In one described implementation, a watermarking system employs chess  
4 spread-spectrum sequences (i.e., "chess watermarks") to improve the balance of  
5 positive and negative chips in the watermarking sequences. The balance is not  
6 imposed in an orderly fashion, which might make the watermark sequence more  
7 easily detectable to an attacker, but in a pseudo-random fashion. In that way, better  
8 sequence balance is achieved while preserving its randomness for an attacker  
9 without knowledge of the keys.

10 In another described implementation, a watermarking system employs an  
11 energy-level trigger to determine whether to skip encoding of a portion of a  
12 watermark within a given time span of an audio clip. If a large discrepancy in  
13 energy levels exists over a given time frame, then the frame is not watermarked, to  
14 avoid audible time-dispersion of artifacts due to spectral modifications (which are  
15 similar to "pre-echo" effects in audio coding). In another described  
16 implementation, a watermarking system begins encoding of a watermark at a  
17 variable position after the beginning of an audio clip.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The same numbers are used throughout the drawings to reference like elements and features.

Fig. 1 is a block diagram of an audio production and distribution system in which a content producer/provider watermarks audio signals and subsequently distributes that watermarked audio stream to a client over a network.

Fig. 2 is a block diagram of a watermarking encoding system implemented, for example, at the content producer/provider.

Fig. 3 is a block diagram of a watermarking detecting unit implemented, for example, at the client.

Figs. 4A-4D show graphs of an audio clip to illustrate blocking and framing of such audio clip.

Fig. 5 illustrate sample blocks and sample frames of an audio clip and further illustrate the encoding of bit values of a watermark within such blocks and frames.

Figs. 6A-6D show redundant encoding of a bit in the blocks of a frame and the effect of implementations of chess watermarking techniques.

Fig. 7 is a flow diagram showing a methodological implementation of chess watermark encoding.

Fig. 8 is a flow diagram showing a methodological implementation of chess watermark decoding.

Fig. 9 shows a plot of a portion of an audio signal to illustrate a large discrepancy of energy level in a block.

1 Fig. 10 is a flow diagram showing a methodological implementation of  
2 improved stealthy audio watermarking with energy-level triggering.

3 Figs. 11A-11C show graphs of an audio clip to illustrate variable starting  
4 positioning for watermark encoding.

5 Fig. 12 is a flow diagram showing a methodological implementation of  
6 improved stealthy audio watermarking with variable starting position.

7 Fig. 13 is an example of a computing operating environment capable of  
8 implementing the improved stealthy audio watermarking.

### 9 10 **DETAILED DESCRIPTION**

11 The following description sets forth a specific embodiment of the  
12 improved stealthy audio watermarking that incorporates elements recited in the  
13 appended claims. This embodiment is described with specificity in order to meet  
14 statutory written description, enablement, and best-mode requirements. However,  
15 the description itself is not intended to limit the scope of this patent. Rather, the  
16 inventors have contemplated that the claimed improved stealthy audio  
17 watermarking might also be embodied in other ways, in conjunction with other  
18 present or future technologies.

### 19 20 **Incorporation by Reference**

21 The following provisional application (from which priority is claimed) is  
22 incorporated by reference herein: U.S. Provisional Patent Application Serial No.  
23 60/143432 entitled "Improved Audio Watermarking" filed on July 13, 1999.  
24  
25

1 In addition, the following co-pending patent applications are incorporated  
2 by reference herein:

- 3 • U.S. Patent Application Serial No. 09/316,899, entitled "Audio  
4 Watermarking with Dual Watermarks" filed on May 22, 1999, and  
5 assigned to the Microsoft Corporation; and
- 6 • U.S. Patent Application Serial No. 09/259,669, entitled "A System  
7 and Method for Producing Modulated Complex Lapped Transforms"  
8 filed on February 26, 1999, and assigned to the Microsoft  
9 Corporation.

10 The following U.S. Patent is incorporated by reference herein: U.S. Patent  
11 No. 6,029,126, entitled "Scalable Audio Coder and Decoder" issued on February  
12 22, 2000, and assigned to the Microsoft Corporation.

### 13 Introduction

14 Described herein are at least three exemplary implementations of improved  
15 stealthy audio watermarking (i.e., "exemplary watermarking"). The first  
16 exemplary watermarking implementation employs chess spread-spectrum  
17 sequences (i.e., "chess watermarks") to improve the short-time statistical balance  
18 of watermark sequences. To detect such a watermark, a watermark detector is  
19 aware that watermarks were encoded using the exemplary chess watermarking.  
20

21 The second exemplary watermarking implementation employs watermark  
22 encoding triggered by the energy level of the signal (i.e., "energy-level trigger").  
23 To detect a watermark, the detector need not be aware that watermarks were  
24 encoded using the exemplary watermarking with energy-level triggering.  
25



1 The third exemplary watermarking implementation employs variable  
2 starting position for watermark encoding (i.e., "variable-starting position"). To  
3 detect a watermark, the detector need not be aware that watermarks were encoded  
4 using the exemplary watermarking with variable start.

5 The exemplary watermarking implementations, described herein, may be at  
6 least implemented by an audio production and distribution system like that shown  
7 in Fig. 1 and by a computing environment like that shown in Fig. 13.

8 The exemplary watermarking implementations, described herein, further  
9 many of the goals of watermarking. They bring one to the gates of the city of "El  
10 Dorado" where an encoded watermark is unalterable, irremovable, and cannot be  
11 de-synced without perceptually and noticeably affecting the audio quality.  
12 Likewise, they bring one within reach of the "Holy Grail" where a malevolent  
13 attacker may know how the watermark is encoded, but still cannot effectively  
14 attack it without perceptually and noticeably affecting the audio quality.

15 A watermark may be generically called an "information pattern of multiple  
16 discrete values" because it is a pattern of binary bits designed to convey  
17 information. A watermark is encoded in a digital audio signal (or clip). In relation  
18 to the watermark, the audio signal is effectively "noise." In general, watermarking  
19 involves hiding the information contained in the watermark within the "noise" of a  
20 digital signal.

## **Audio Production and Distribution System Employing Watermarks**

Fig. 1 shows an audio production and distribution system 20 having a content producer/provider 22 that produces original musical content and distributes the musical content over a network 24 to a client 26. The content producer/provider 22 has a content storage 30 to store digital audio streams of original musical content. The content producer 22 has a watermark encoding system 32 to sign the audio data stream with a watermark that uniquely identifies the content as original. The watermark encoding system 32 may be implemented as a standalone process or incorporated into other applications or an operating system.

A watermark is an array of bits generated using a cryptographically secure pseudo-random bit generator and a new error correction encoder. The pseudo-uniqueness of each watermark is provided by initiating the bit generator with a key unique to each audio content publisher. The watermark is embedded into a digital audio signal by altering its frequency magnitudes such that the perceptual audio characteristics of the original recording are preserved. Each magnitude in the frequency spectrum is altered according to the appropriate bit in the watermark.

The watermark encoding system 32 applies the watermark to an audio signal from the content storage 30. Typically, the watermark identifies the content producer 22, providing a signature that is embedded in the audio signal and cannot be removed. The watermark is designed to survive all typical kinds of processing, including compression, equalization, D/A and A/D conversion, recording on analog tape, and so forth. It is also designed to survive malicious attacks that

1 attempt to remove the watermark from the signal, including changes in time and  
2 frequency scales, pitch shifting, and cut/paste editing.

3 The content producer/provider 22 has a distribution server 34 that streams  
4 the watermarked audio content over the network 24 (e.g., the Internet). An audio  
5 stream with a watermark embedded therein represents to a recipient that the stream  
6 is being distributed in accordance with the copyright authority of the content  
7 producer/provider 22. The server 34 may further compress and/or encrypt the  
8 content conventional compression and encryption techniques prior to distributing  
9 the content over the network 24.

10 The client 26 is equipped with a processor 40, a memory 42, and one or  
11 more media output devices 44. The processor 40 runs various tools to process the  
12 audio stream, such as tools to decompress the stream, decrypt the data, filter the  
13 content, and/or apply audio controls (tone, volume, etc.). The memory 42 stores  
14 an operating system 50 (such as a Microsoft® Windows 2000® operating system),  
15 which executes on the processor. The client 26 may be embodied in a many  
16 different ways, including a computer, a handheld entertainment device, a set-top  
17 box, a television, an audio appliance, and so forth.

18 The operating system 50 implements a client-side watermark detecting  
19 system 52 to detect watermarks in the audio stream and a media audio player 54 to  
20 facilitate play of the audio content through the media output device(s) 44 (e.g.,  
21 sound card, speakers, etc.). If the watermark is present, the client can identify its  
22 copyright and other associated information.

23 The operating system 50 and/or processor 40 may be configured to enforce  
24 certain rules imposed by the content producer/provider (or copyright owner). For  
25

1 instance, the operating system and/or processor may be configured to reject fake  
2 or copied content that does not possess a valid watermark. In another example, the  
3 system could play unverified content with a reduced level of fidelity.  
4

### 5 **Watermark Insertion and Detection**

6 Some of the basal details of watermark insertion and detection are  
7 thoroughly described in U.S. Patent Application Serial No. 09/316,899, entitled  
8 "Audio Watermarking with Dual Watermarks" filed on May 22, 1999 (which, as  
9 indicated above, is incorporated by reference, herein).  
10

11 In general, Fig. 2 shows a watermark encoding system 100 (or simply  
12 "watermark encoder") that may be implemented at a content provider/producer to  
13 encode the audio signal with a watermark. The watermark encoding system 100  
14 has a converter 110 to convert an audio signal into frequency-domain magnitude  
15 and phase components. It may also include an energy-level trigger 112 to  
16 determine whether the energy level across a portion of the signal warrants  
17 encoding of the watermark in that portion.

18 The watermark encoding system 100 also has a pattern generator 114 to  
19 generate the watermark and a watermark insertion unit (WIU) 116 to insert the  
20 watermark into the signal. The pattern generator typically includes a  
21 pseudorandom number generator (PRNG) to generate a watermark based upon a  
22 watermark key. The WIU 116 receives magnitude components from the converter  
23 110, a triggering signal from the trigger 112, and the watermark from the pattern  
24 generator 114. The trigger 112 generates a YES/NO signal to indicate to a  
25

1 watermark insertion unit whether to encode a watermark in a specified portion of a  
2 signal.

3 The watermark encoding system 100 has a deconverter 118 to convert the  
4 audio signal back into the time domain. Pseudorandom number generator (PRNG)  
5 120 is employed to implement the exemplary watermarking, but its role is  
6 explained later.

7 In general, Fig. 3 shows a watermark detecting system 130 (or simply  
8 "watermark detector") that may be implemented at a client that plays the audio  
9 clip (containing the audio signal). In addition, it may be implemented in an audio  
10 management and distribution subsystem (for example, in an application that  
11 downloads music clips from the Internet and uploads them to portable devices).

12 The watermark detecting system 130 has a converter 140, a mask processor  
13 142, and a watermark pattern generator 144. The converter 140 converts an audio  
14 signal that is suspected to include a watermark. It converts the signal into its  
15 frequency-domain magnitudes. The mask processor 142 determines the hearing  
16 threshold for frequency-domain magnitude components. The pattern generator  
17 144 generates a comparison watermark based upon the same watermark key as  
18 used by the encoder. The pattern generator 144 typically includes a pseudorandom  
19 number generator (PRNG) to generate the comparison watermark based upon a  
20 watermark key.

21 The watermark detecting system 130 is also equipped with a watermark  
22 detector 146 that locates a watermark in the audio clip. The watermark detector  
23 146 determines which block interval of the watermarked audio signal contains a  
24 watermark pattern and whether that discovered watermark pattern matches the  
25

1 comparison watermark generated by the pattern generator 144. Pseudorandom  
2 number generator (PRNG) 150 is employed to implement the exemplary  
3 watermarking, but its role is explained later.  
4

## 5 **Blocks and Frames**

6 During the encoding, the original audio signal is processed into equally  
7 sized, overlapping, time-domain blocks. Each of these blocks is the same length  
8 of time. For example, one second, two seconds, 50 milliseconds, and the like. In  
9 addition, these blocks overlap equally so that half of each block (except the first  
10 and last) is duplicated in an adjacent block.

11 For example, suppose that an audio clip is divided into overlapping, two-  
12 second long, time-domain blocks. This means that each block has a one second  
13 overlap with its neighbors. If the clip were about 3.5 minutes long, then there  
14 would be about 210 blocks.

15 Fig. 4A shows a graph 300 of an audio signal in the time domain. Time  
16 advances from left to right. Fig. 4B shows a graph 320 of the same audio signal  
17 sampled over the same time period. Fig. 4B includes a block 322 representing a  
18 first of equally spaced, overlapping, time-domain blocks.

19 Each block is transformed by a MCLT (modulated complex lapped  
20 transform) to the frequency domain. This produces a vector having a defined  
21 number of magnitude and phase components. The magnitude is measured in a  
22 logarithmic scale, in decibels (dB).

23 Fig. 4C shows a graph 340 of the same audio signal sampled over the same  
24 time period. In Fig. 4C, there is a set 350 of five adjacent blocks 352-359. The  
25

1 blocks represent equally spaced, overlapping, time-domain blocks. (For  
2 simplicity, the overlapping nature of the blocks is not shown.) The set 350 is  
3 called a "frame." A frame may include any given number of blocks.

4 Fig. 4D shows a graph 360 of the same audio signal sampled over the same  
5 time period. In Fig. 4D, there are three frames 370, 380, and 390. Each frame has  
6 five adjacent blocks. The blocks represent equally spaced, overlapping, time-  
7 domain blocks. (For simplicity, the overlapping nature of the blocks is not  
8 shown.)

9 Fig. 5 shows a graph 400 of the same audio clip of Figs. 4A-4B, but this  
10 graph does not show the clip in the time domain. Rather, it shows a graph in the  
11 frequency-domain for each overlapping, time-domain block (like blocks 352-359  
12 in Fig. 4C). Time advances from left to right. This is from the beginning of the  
13 audio clip to the end. Frequency increases from bottom to top. From zero to a  
14 maximum frequency of interest ("MaxFreq").

15 In Fig. 5, each of blocks 412a-g contain a frequency-domain graph for its  
16 time blocks. Horizontal hash marks, like mark 414, represent the magnitude of a  
17 given frequency range. Each watermark chip is encoded in multiple frequency  
18 subbands in a range from "SubBand<sub>max</sub>" line and "SubBand<sub>min</sub>" line as shown in  
19 Fig. 5.

20 A given number of blocks (such as blocks 412a-g) form a "frame" (such as  
21 frame 410). Each frame includes the same number of blocks. In Fig. 5, frames  
22 420, 430, and 440 includes the same number of blocks.

## Encoding Bits of a Watermark

A watermark is composed of a given number of bits (such as eighty bits). The bits of a watermark are encoded by slightly increasing and decreasing the magnitude of frequencies within a block. This slight change is plus or minus  $Q$  decibel (dB), where  $Q$  is set to 1 for example. These frequency changes are not heard because they are too small. Again, these frequency magnitudes are represented by horizontal hash marks, like mark 414.

More specifically, only the frequencies between the  $\text{SubBand}_{\text{max}}$  and  $\text{SubBand}_{\text{min}}$  lines are modified to encode a bit of the watermark.

## Redundancy Encoding

Successive Redundancy of Full Watermark. Using the exemplary watermarking, successive bits are stored in successive frames. One bit is encoded in each frame. For example, suppose the watermark is eighty bits long. The first three bits of the watermark in this example is "101" and its last bit is "0". Also, suppose that frame 410 is frame one, frame 420 is frame two, and so forth until frame 440 is frame eighty.

In this example, frame 410 will have the first bit of the watermark encoded therein. That bit is "1" and is represented by indicator 450. Frame 420 will have the second bit of the watermark encoded therein. That bit is "0" and is represented by indicator 452. Frame 430 will have the third bit of the watermark encoded therein. That bit is "1" and is represented by indicator 454. Finally, frame 440 will have the last bit of the watermark encoded therein. That bit is "0" and is represented by indicator 456.



1 Typically, the full audio clip in which the watermark is being encoded is  
2 longer than time elapsed for the eighty frames. Therefore, this process is repeated  
3 until the end of the audio clip. In one implementation, it was determined that  
4 approximately eleven seconds was required to encode a watermark. Thus, in a  
5 four-minute clip, the watermark will be encoded approximately twenty-one times  
6 in successive sets of eighty frames. That allows the watermark to be detected even  
7 by looking only at a small portion of the audio clip.

8 Redundancy within a Frame. As described above, each frame has one bit  
9 of the watermark encoded therein. That one bit is encoded in each block of a  
10 frame. This means that within each block in a frame is encoded the exact same bit.  
11 For example, indicator 450 of Fig. 5 shows that each block in frame 410 has a bit  
12 value of "1" encoded therein.

13 When a bit of a watermark is detected from an audio clip, the detector reads  
14 the bit from the block in the middle of frame. In frame 410 of Fig. 5, the middle  
15 block is block 412d.

16 The redundancy within a frame is designed to thwart malevolent  
17 desynchronization attacks in the time-domain. In other words, it lessens the effect  
18 of time-shifting the audio clip. Since it reads what it believes to be the middle  
19 block of a frame, it will still read the correct bit value even if the clips is shifted  
20 over an amount of time equal to about half of a frame.

## 21 Redundancy Problem

22 The relative quantity of bits in a normal audio clip is roughly balanced  
23 between "1's" and "0's." Likewise, the distribution of bits (i.e., "1's" and "0's") in  
24  
25

004460-0400

1 a normal audio clip is roughly evenly distributed throughout the clips and over  
2 most any given portion of such clip. This assumption is correctly made because  
3 an audio clip is a digital representation of an analog recording (such as music and  
4 talking).

5 In generally, entropy is a quantitative measure of uncertainty. Entropy may  
6 also be defined as a measure of the disorder or randomness in a closed system.  
7 Therefore, the bits of an original audio signal appear to be disordered and random

8 Anything that is digitally encoded typically has a pattern so that it may be  
9 recognized by a detector. Since a purposeful action places some order upon a  
10 signal, the bits in such a signal no longer appear random and disordered. Digital  
11 pirates know this. They also know that the digitally encoded pattern is typically  
12 repeated in an audio clip.

13 Therefore, one type of malevolent attack is to search an audio clip for  
14 patterns, particularly repeated patterns. Of course, redundancy itself is a pattern.  
15 Once a pirate finds such a pattern, he or she may attempt to remove it, change it,  
16 or scramble it. This is done so that the detector either does not find a watermark  
17 or misidentifies a watermark.

18 However, as discussed above the watermark itself is repeated throughout  
19 the audio clip and each bit of the watermark is repeated in a series of blocks within  
20 a frame. With such pattern redundancy, there is an increased danger of detection  
21 by a malicious attacker.

## Entropy-Balancing of Watermarks

Using the exemplary chess watermarking, the patterns are effectively “hidden” by further encoding (re-encoded) them to obscure the patterns. The patterns may still be found by the detector because it knows how they were re-encoded and thus, the detector “de-re-encodes” before it detects the watermark as normal.

Generically, this technique may be referred to as “entropy-balancing” of patterns, in particular, watermarks. After such entropy-balancing, the bits in a pattern no longer appear to be organized, ordered, and non-random. Rather, the bits of the pattern appear to be disorganized, unordered, and random. Thus, the entropy of the bits of the pattern appears to be balanced. A watermark generated by the exemplary watermarking techniques is a “watermark with balanced entropy.”

The exemplary watermarking has the ability to tell the value of the next bit generated by a pseudorandom number generator (PRNG) under any previous history of generated bits.

Figs. 6A-6B illustrate entropy-balancing of blocks in a frame. Since each frame represents one bit of a watermark, the entropy-balancing of each frame effectively entropy-balances the entire encoded watermark. To further hide the watermark pattern, each encoded watermark is entropy-balanced independently from each of the other watermarks. Therefore, the same entropy-balanced watermark is not simply repeated.

Fig. 6A shows a frame 470 of ten blocks. Although a frame typically has an odd number of blocks, this is provided for illustration purposes only. The frame

1 represents one bit of an encoded watermark. That bit is "1." As shown in Fig. 6A,  
2 each block is encoded with that bit. Thus, the blocks of frame 470 have encoded  
3 therein the same original bit.

4 However, this is a clear pattern of repeated "1's." While such a pattern may  
5 appear naturally, it is not natural for a set of exactly ten homogeneous bits (either  
6 all "0's" or all "1's") to appear one after the other. This pattern is not entropy-  
7 balanced. A digital bandit may easily discover such pattern.

8 Fig. 6B shows the results of "absolute-chessboarding" the pattern in frame  
9 472. Absolute chessboarding is one option for hiding a pattern. Absolute  
10 chessboarding changes every other block. For example, the pattern of the original  
11 frame 470 is "1111111111" is absolutely chessboarded into alternating "1's" and  
12 "0's" beginning with "1." In Fig. 6B, the absolutely chessboarded pattern of frame  
13 472 is now "1010101010." In this example, every other block is reversed starting  
14 with the second block. Alternatively, such reversal may begin with the first block.

15 This absolute-chessboard pattern represents a perfect distribution of bits  
16 within a frame. The "1's" and "0's" are perfectly distributed within a frame.  
17 However, this distribution is probably too perfect. Each frame begins with either a  
18 "1" or a "0" and the remainder of the frame has perfect bit distribution. Thus, this  
19 repeating pattern is discoverable.

20 Although an absolutely chessboarded pattern (such as that of frame 472)  
21 may be more difficult to discover than a solid pattern (such as that of frame 470),  
22 it is still a redundant pattern of alternating bits every ten blocks. An absolutely  
23 chessboarded pattern is still a recognizable pattern; thus, it is not entropy-  
24 balanced.  
25

## Pseudorandom-Chessboarding

Figs. 6C and 6D illustrate the results of pseudorandom-chessboarding the pattern in frame 472. When compared to frames 470 and 472, the patterns of frames 474 and 476 appear to be random and non-ordered.

To generate this pseudorandom-chessboard pattern, each block is processed by a pseudorandom number generator (PRNG) so that the resulting pattern is entropy-balanced.

When encoding a bit of the watermark into a frame, the encoder (within the watermark insertion unit 116 of Fig. 2) processes the bit of each block before inserting it. The PRNG (such as PRNG 120 in Fig. 2) gives a result (typically between 0 and 1), which is compared to a threshold value. If that threshold value is 0.5, then the result is an absolute chessboarded pattern. If that threshold value is 1 or 0, then the result is the same or the reverse of the original pattern. Therefore, the threshold value is typically a value that does not approach 0.5, 0, or 1. In the exemplary chess watermarking, the threshold value is typically 0.65-0.85.

The watermark encoder and detector use the same PRNG engine (such as PRNG 120 in Fig. 2 and PRNG 150 in Fig. 3) and the same key. This key may be the watermark key and typically accompanies the digital audio file in a cryptographic manner. Since the encoder and detector use the same PRNG engine and key, the resulting pattern of determining when to alternate bits is the same for both the encoder and detector. Alternatively, a look-up table may be used to achieve the same results.

Fig. 6D illustrates the preferred results of such pseudorandom-chessboarding. A frame (and consequently the repeated watermark in the entire

audio clip) is the most entropy-balanced when the pattern approaches the ideal distribution of alternating "1's" and "0's," but it does not reach it. Frame 476 has this entropy-balanced pattern: "1010010101." Although not shown, the next frame may have this pattern: "0110101011." With similar minor variation from the ideal distribution in each frame of the watermark, the pattern will be nearly impossible to discover without knowledge of the PRNG engine and the key (or of the look-up table).

Although Fig. 6D does illustrate the preferred results of such pseudorandom-chessboarding where the pattern approaches the ideal distribution. Those of ordinary skill in the art understand and appreciate that "entropy-balanced" refers to an apparently random and disorderly pattern between the solid pattern of frame 470 and the ideal distribution of frame 472.

A watermark that results from chessboarding may be called a "chess watermark" because it appears to be a chessboard if graphed two-dimensionally. Likewise, the act of processing a watermark in this fashion may be called "chess watermarking" or "chessboarding."

### **Methodological Implementation of Exemplary Chess Watermark Encoding**

Fig. 7 shows a methodological implementation of the exemplary chess watermark encoding. At 500, an original audio signal (such as from an audio clip) is preprocessed. The effective result of such preprocessing is to produce blocks and frames as described above.

Furthermore, such signal preprocessing is generally described above in reference to the watermark encoding system of Fig. 2. It is also described in more

1 detail in co-pending patent application: U.S. Patent Application Serial No.  
2 09/316,899, entitled "Audio Watermarking with Dual Watermarks" filed on May  
3 22, 1999.

4 At 502, the watermark encoder generates a watermark in accordance with  
5 watermark generation described above and in the "Audio Watermarking with Dual  
6 Watermarks" co-pending application.

7 At 504, the blocks of the audio signal and the watermark are provided to a  
8 watermark insertion unit (such as unit 116 in Fig. 2). At 506, before the bits of the  
9 watermark are inserted into the signal, they are "chessboarded." For example, a  
10 PRNG (such as PRNG 120 in Fig. 2) generates a pseudorandom number to  
11 determine whether to switch a bit.

12 At 508 in Fig. 7, the resulting chess watermark is inserted into the audio  
13 signal. At 510, this process ends.

14 The following are two examples of pseudocode that may be used to  
15 implement exemplary chess watermark encoding:

16  
17 Example One: Pseudorandom-Chessboarding using a PRNG

18 INPUT={SECRET\_KEY}  
19 OUTPUT={WATERMARK(S,K) OF LENGTH K={2,4,6,8} TIME BLOCKS IN S  
20 SUBBANDS}  
21  
22 CREATE\_CHESS\_WATERMARK (SECRET\_KEY) {  
23     INITIATE PSEUDO\_RANDOM\_NUMBER\_GENERATOR RANDOM with SECRET\_KEY  
24     { PROBABILISTIC AUTOMATON:  
25         STATE[0] = OUTPUT ZERO  
26         STATE[1] = OUTPUT ONE  
27         CHANGE\_STATE(STATE(Y)): GENERATE p=RANDOM(): if p>THRESHOLD  
28         goto STATE(not(Y)) else stay in STATE(Y)  
29         //common watermarks (absolute chessboard) are generated  
30         //with THRESHOLD=1/2.Using this automaton, it enforces  
31         //arbitrarily the probability that bits in consecutive  
32         //blocks in the same subband are toggled. TYPICALLY  
33         //RANGE OF OPERATION FOR THRESHOLD IS 0.65-0.8.  
34     }  
35     FOREACH SUBBAND s

```

1         FOR k=1:K
2             WATERMARK(s,k)=Y of STATE(Y) - get the bit
3             corresponding to the state
4             CHANGE_STATE(STATE(Y)) - change the state pseudo-
5             randomly
6         ENDFOR
7     ENDFOREACH
8 }
9 =====

```

#### Example Two: Pseudorandom-Chessboarding using a Look-up Table

```

6 -----
7 CREATE_CHESS_WATERMARK (SECRET_KEY) {
8     // TWO POSSIBLE WATERMARKS FOR TWO BIT LONG WATERMARK SEQUENCE
9     LOOKUP_TABLE2X2 [2][2] = { 1, 0,
10                                0, 1};
11
12     // SIX POSSIBLE WATERMARKS FOR FOUR BITS LONG WATERMARK SEQUENCE
13     LOOKUP_TABLE6X4 [6][4] = { 1, 1, 0, 0,
14                                1, 0, 1, 0,
15                                1, 0, 0, 1,
16                                0, 0, 1, 1,
17                                0, 1, 0, 1,
18                                0, 1, 1, 0};
19
20     // CREATION OF WATERMARK TABLES OF LENGHT 6 AND 8 BITS
21     // BOTH TABLES ARE CREATED AS A SET OF ALL POSSIBLE
22     // CONCATENATIONS OF LOWER ORDER TABLES
23     for (i = 0; i < 6; i++)
24         for (j = 0; j < 6; j++)
25             for (k = 0; k < 8; k++)
26                 if (k < 4) LOOKUP_TABLE36x8[i*6+j][k] =
27                     LOOKUP_TABLE6X4[i][k];
28                 else LOOKUP_TABLE36x8[i*6+j][k] =
29                     LOOKUP_TABLE6X4[j][k-4];
30     for (i = 0; i < 6; i++)
31         for (j = 0; j < 2; j++)
32             for (k = 0; k < 6; k++)
33                 if (k < 4) LOOKUP_TABLE12X6[i*2+j][k] =
34                     LOOKUP_TABLE6X4[i][k];
35                 else LOOKUP_TABLE12X6[i*2+j][k] =
36                     LOOKUP_TABLE2X2[j][k-4];
37
38     // A WATERMARK WATERMARK(S,K) OF LENGTH K={2,4,6,8} IN S SUBBANDS
39     // IS CREATED IN THE FOLLOWING WAY:
40     INITIATE PSEUDO_RANDOM_NUMBER_GENERATOR RANDOM with SECRET_KEY
41     FOREACH SUBBAND s IN S
42         WATERMARK(s, K) = LOOKUP_TABLE?XK[RANDOM()][ ]
43     ENDFOREACH
44 }
45 =====

```



## Methodological Implementation of Exemplary Chess Watermark Detecting

Fig. 8 shows a methodological implementation of the exemplary chess watermark detecting. At 520, a watermarked audio signal (such as from an audio clip) is preprocessed. The effective result of such preprocessing is to produce blocks and frames.

Furthermore, such signal preprocessing is generally described above in reference to the watermark detecting system of Fig. 3. It is also described in more detail in co-pending patent application: U.S. Patent Application Serial No. 09/316,899, entitled "Audio Watermarking with Dual Watermarks" filed on May 22, 1999.

At 522, the watermark detector generates a comparison watermark in accordance with watermark generation described above and in the "Audio Watermarking with Dual Watermarks" co-pending application. This comparison watermark is generated using the same key as the original watermark. Therefore, they are identical.

At 524, the blocks of the audio signal and the comparison watermark are provided to a watermark detector unit (such as unit 146 in Fig. 3). At 526, the watermark is detected from the audio signal.

At 528 in Fig. 8, the detected watermark is "un-chessboarded." This means that the same technique used to generate the chessboard pattern is used to return the bits of the watermark back to their original pattern.

For example, a PRNG (such as PRNG 150 in Fig. 3) generates pseudorandom number to determine whether to switch a bit. It uses the same

1 engine as the PRNG of the encoder. It also uses the same key as the PRNG of the  
2 encoder; therefore, the resulting pseudorandom pattern is the same.

3 At 530, this process ends. Typically, the detector will generate a result that  
4 indicates whether a watermark is present in the audio signal.

### 6 **Energy-Level Triggering**

7 As mentioned previously, the inserting of a bit value (one or zero) of the  
8 watermark involves slight modification of frequency magnitudes in the frequency  
9 domain of a block. That slight modification is an addition or subtraction of  
10 typically one dB. Since dBs are on a logarithmic scale, this modification is  
11 difficult to detect. That is, unless there is a large discrepancy (e.g. a factor of three  
12 or more) in the energy levels across the time blocks in which the bit is being  
13 written.

14 Because changes in the magnitude frequency components tend to spread in  
15 time, a change that is small with respect to a large energy portion of a block may  
16 be large with respect to a small energy portion. That could make the change  
17 detectable and help a pirate in a malicious attack attempt. Also, it could lead to  
18 audible distortions.

19 Fig. 9 shows a graph 550 of partial signal 555 and illustrates this issue.  
20 Time advances from left to right on the graph 550. Up and down illustrates the  
21 amplitude (i.e., magnitude, energy level) of the signal 555.

22 A fragment 562 of such signal 555 is shown in Block A 560. The energy  
23 level across the frequency spectrum in that fragmented signal 562 is roughly  
24 similar.

1 Compare that to fragments 572 and 574 of signal 555 as shown in Block B  
2 570. The energy level of fragmented signal 572 is dramatically different from that  
3 of fragmented signal 574. Fragmented signal 572 has a very low energy level, but  
4 fragmented signal 574 has a very high energy level.

5 In the exemplary watermark encoder using an energy-level trigger, the  
6 encoder detects blocks having a large discrepancy in energy level and skips them—,  
7 i.e., it simply does not insert the watermark in that block.

8 The detector need not know that the encoder skipped a block. It does not  
9 matter if a few watermarks were not encoded because of energy-level triggering.  
10 There are plenty of other watermarked blocks in the audio clip to ensure correct  
11 operation of the correlation-based detector (as described in more detail in co-  
12 pending patent application: U.S. Patent Application Serial No. 09/316,899, entitled  
13 “Audio Watermarking with Dual Watermarks” filed on May 22, 1999).

## 14 Methodological Implementation of

### 15 Exemplary Watermark Encoding with Energy-Level Triggering

16 Fig. 10 shows a methodological implementation of the exemplary  
17 watermark encoding with energy-level triggering. At 600, before inserting a bit of  
18 a watermark into a block of a frame, the encoder (such as the one shown in Fig. 2)  
19 analyzes the energy level across the spectrum within such block.  
20

21 At 602, the encoder determines if there is a large discrepancy in energy  
22 levels of the analyzed block. If not, a watermark bit is inserted into the block at  
23 604. After that, the process continues at block 606. At 606, the normal encoding  
24 process continues for remaining blocks in audio signal. If there is a large  
25

1 discrepancy in energy levels, then that block is skipped and the process jumps to  
2 block 606.

3 The following is an example of pseudocode that may be used to implement  
4 exemplary watermark encoding with energy-level triggering:

5  
6 Example: Watermark Encoding using Energy-Level Triggering

7 -----  
8 ROUTINE TO\_WATERMARK WHICH CHECKS for pre-echo PROBLEMS  
9 -----

10 INPUT=BLOCK OF FREQUENCY MAGNITUDES {BLOCK}

11 OUTPUT=DECISION TO WATERMARK  
12 -----

13 {YES,NO} = TO\_WATERMARK(BLOCK) {

14 SCALE = #(FREQUENCY MAGNITUDES PER FREQUENCY SPECTRUM BLOCK) /  
15 #(SUBBAND PARTITIONS)

16 max\_energy = 0

17 min\_energy = LARGEST\_POSSIBLE\_NUMBER

18 FOREACH SUBBAND PARTITION sp {

19 ENERGY = COMPUTE NORMALIZED SUM OF ENERGY OF ALL FREQUENCY  
20 MAGNITUDES IN SUBBAND sp

21 if (ENERGY > max\_energy) max\_energy = ENERGY

22 if (ENERGY < min\_energy) min\_energy = ENERGY

23 }

24 if (max\_energy/min\_energy < ERLIM) return(YES); else return(NO)

25 }

ERLIM is established empirically and for example for 8 subbands,  
ERLIM is used in the range of 100 and 200.

=====

17  
18 Variable Starting Position of Watermark

19 Averaging Attacks. An averaging attack is another form of malevolent  
20 attack. Typically, a music publisher often uses the same key to generate their  
21 watermark in each published audio clip. This key is designed to identify the  
22 publisher.

23 If a digital audio rustler processes a collection of audio clips that were  
24 encoded using a common key, then the rustler may “average” out all of the music  
25

1 (which is effectively noise) to find a commonly encoded watermark in each clip.  
2 Once this watermark is found, it can be removed or changed.

3 Variable Starting Position. An averaging attack may be thwarted by  
4 varying the starting point to initiate encoding of the watermark. Since the  
5 watermarks in each of a collection of common-key clips are time shifted relative  
6 each other, averaging them provides no useful information.

7 Figs. 11A-C illustrate this variable-starting-position encoding. Fig. 11A  
8 shows a time graph 700 of an audio clip. The clip starts at the point indicated by  
9 arrow 702.

10 Fig. 11B shows a graph 720 of the same audio clip of Fig. 11A. The clip  
11 starts at the point indicated by arrow 722. Fig. 11B also includes a representation  
12 of a simplified watermark at 730. The beginning of the encoded watermark begins  
13 at the point indicated by arrow 732. Like what is shown in Fig. 11B, the  
14 beginning 732 of watermark encoding conventionally coincides with the  
15 beginning 722 of the audio clip.

16 Fig. 11C illustrates a graph 740 of the same audio clip of Figs. 11A and  
17 11B. The clip starts at the point indicated by arrow 742. Fig. 11C also includes a  
18 representation of a simplified watermark at 750. The beginning of the encoded  
19 watermark begins at the point indicated by arrow 752. Unlike conventional  
20 approaches, the beginning 752 of watermark encoding occurs after the beginning  
21 742 of the audio clip.

22 How far after? In one example, a PRNG pseudorandomly selects an  
23 amount of time after the beginning of the clip to begin encoding the watermark.  
24 This may use a PRNG like that used for the exemplary chess watermark encoding.  
25

1 If the PRNG uses the same key when the audio clip is being detected, then  
2 detecting will begin at the correct moment in the clip.

### 3 4 **Methodological Implementation of**

#### 5 **Exemplary Watermarking with Variable Starting Position**

6 Fig. 12 shows a methodological implementation of the exemplary  
7 watermark encoding with variable starting position. At 800, the detector initiates  
8 preprocessing of an original audio signal. However, watermark encoding does not  
9 yet begin.

10 Such signal preprocessing is generally described above in reference to the  
11 watermark encoding system of Fig. 2. It is also described in more detail in co-  
12 pending patent application: U.S. Patent Application Serial No. 09/316,899, entitled  
13 "Audio Watermarking with Dual Watermarks" filed on May 22, 1999.

14 At 802, the encoder waits for a pseudorandom amount of time. This period  
15 of time may be determined by a PRNG using a given key. The PRNG of the  
16 detector uses the same key; thus, it begins detecting at the correct moment in the  
17 clip.

18 At 804, the encoder initiates insertion of the watermark into the audio  
19 signal. At 806, the normal encoding process continues for the remainder of the  
20 audio signal.

### 21 22 **Exemplary Computing Environment**

23 Fig. 13 illustrates an example of a suitable computing environment 920 on  
24 which the exemplary watermarking may be implemented.

1 Exemplary computing environment 920 is only one example of a suitable  
2 computing environment and is not intended to suggest any limitation as to the  
3 scope of use or functionality of the exemplary watermarking. Neither should the  
4 computing environment 920 be interpreted as having any dependency or  
5 requirement relating to any one or combination of components illustrated in the  
6 exemplary computing environment 920.

7 The exemplary watermarking is operational with numerous other general  
8 purpose or special purpose computing system environments or configurations.  
9 Examples of well known computing systems, environments, and/or configurations  
10 that may be suitable for use with the exemplary watermarking include, but are not  
11 limited to, personal computers, server computers, thin clients, thick clients, hand-  
12 held or laptop devices, multiprocessor systems, microprocessor-based systems, set  
13 top boxes, programmable consumer electronics, network PCs, minicomputers,  
14 mainframe computers, distributed computing environments that include any of the  
15 above systems or devices, and the like.

16 The exemplary watermarking may be described in the general context of  
17 computer-executable instructions, such as program modules, being executed by a  
18 computer. Generally, program modules include routines, programs, objects,  
19 components, data structures, etc. that perform particular tasks or implement  
20 particular abstract data types. The exemplary watermarking may also be practiced  
21 in distributed computing environments where tasks are performed by remote  
22 processing devices that are linked through a communications network. In a  
23 distributed computing environment, program modules may be located in both local  
24 and remote computer storage media including memory storage devices.  
25

1 As shown in Fig. 13, the computing environment 920 includes a general-  
2 purpose computing device in the form of a computer 930. The components of  
3 computer 920 may include, by are not limited to, one or more processors or  
4 processing units 932, a system memory 934, and a bus 936 that couples various  
5 system components including the system memory 934 to the processor 932.

6 Bus 936 represents one or more of any of several types of bus structures,  
7 including a memory bus or memory controller, a peripheral bus, an accelerated  
8 graphics port, and a processor or local bus using any of a variety of bus  
9 architectures. By way of example, and not limitation, such architectures include  
10 Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA)  
11 bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA)  
12 local bus, and Peripheral Component Interconnects (PCI) buss also known as  
13 Mezzanine bus.

14 Computer 930 typically includes a variety of computer readable media.  
15 Such media may be any available media that is accessible by computer 930, and it  
16 includes both volatile and non-volatile media, removable and non-removable  
17 media.

18 In Fig. 13, the system memory includes computer readable media in the  
19 form of volatile, such as random access memory (RAM) 940, and/or non-volatile  
20 memory, such as read only memory (ROM) 938. A basic input/output system  
21 (BIOS) 942, containing the basic routines that help to transfer information  
22 between elements within computer 930, such as during start-up, is stored in ROM  
23 938. RAM 940 typically contains data and/or program modules that are  
24 immediately accessible to and/or presently be operated on by processor 932.  
25



Computer 930 may further include other removable/non-removable, volatile/non-volatile computer storage media. By way of example only, Fig. 13 illustrates a hard disk drive 944 for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"), a magnetic disk drive 946 for reading from and writing to a removable, non-volatile magnetic disk 948 (e.g., a "floppy disk"), and an optical disk drive 950 for reading from or writing to a removable, non-volatile optical disk 952 such as a CD-ROM, DVD-ROM or other optical media. The hard disk drive 944, magnetic disk drive 946, and optical disk drive 950 are each connected to bus 936 by one or more interfaces 954.

The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules, and other data for computer 930. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 948 and a removable optical disk 952, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 948, optical disk 952, ROM 938, or RAM 940, including, by way of example, and not limitation, an operating system 958, one or more application programs 960, other program modules 962, and program data 964.

1 A user may enter commands and information into computer 930 through  
2 input devices such as keyboard 966 and pointing device 968 (such as a "mouse").  
3 Other input devices (not shown) may include a microphone, joystick, game pad,  
4 satellite dish, serial port, scanner, or the like. These and other input devices are  
5 connected to the processing unit 932 through an user input interface 970 that is  
6 coupled to bus 936, but may be connected by other interface and bus structures,  
7 such as a parallel port, game port, or a universal serial bus (USB).

8 A monitor 972 or other type of display device is also connected to bus 936  
9 via an interface, such as a video adapter 974. In addition to the monitor, personal  
10 computers typically include other peripheral output devices (not shown), such as  
11 speakers and printers, which may be connected through output peripheral interface  
12 975.

13 Computer 930 may operate in a networked environment using logical  
14 connections to one or more remote computers, such as a remote computer 982.  
15 Remote computer 982 may include many or all of the elements and features  
16 described herein relative to computer 930.

17 Logical connections shown in Fig. 13 are a local area network (LAN) 977  
18 and a general wide area network (WAN) 979. Such networking environments are  
19 commonplace in offices, enterprise-wide computer networks, intranets, and the  
20 Internet.

21 When used in a LAN networking environment, the computer 930 is  
22 connected to LAN 977 network interface or adapter 986. When used in a WAN  
23 networking environment, the computer typically includes a modem 978 or other  
24 means for establishing communications over the WAN 979. The modem 978,  
25

1 which may be internal or external, may be connected to the system bus 936 via the  
2 user input interface 970, or other appropriate mechanism.

3 Depicted in Fig. 13, is a specific implementation of a WAN via the Internet.  
4 Over the Internet, computer 930 typically includes a modem 978 or other means  
5 for establishing communications over the Internet 980. Modem 978, which may  
6 be internal or external, is connected to bus 936 via interface 970.

7 In a networked environment, program modules depicted relative to the  
8 personal computer 930, or portions thereof, may be stored in a remote memory  
9 storage device. By way of example, and not limitation, Fig. 13 illustrates remote  
10 application programs 989 as residing on a memory device of remote computer  
11 982. It will be appreciated that the network connections shown and described are  
12 exemplary and other means of establishing a communications link between the  
13 computers may be used.

### 14 Exemplary Operating Environment

15 Fig. 13 illustrates an example of a suitable operating environment 920 in  
16 which the exemplary watermarking may be implemented. Specifically, the  
17 exemplary watermarking is implemented by any program 960-962 or operating  
18 system 958 in Fig. 13.

19 The operating environment is only an example of a suitable operating  
20 environment and is not intended to suggest any limitation as to the scope of use of  
21 functionality of the exemplary watermarking described herein. Other well known  
22 computing systems, environments, and/or configurations that may be suitable for  
23 use with the exemplary watermarking include, but are not limited to, personal  
24  
25

1 computers, server computers, hand-held or laptop devices, multiprocessor systems,  
2 microprocessor-based systems, programmable consumer electronics, wireless  
3 communications equipment, network PCs, minicomputers, mainframe computers,  
4 distributed computing environments that include any of the above systems or  
5 devices, and the like.

### 6 7 **Computer-Executable Instructions**

8 An implementation of the exemplary watermarking may be described in the  
9 general context of computer-executable instructions, such as program modules,  
10 executed by one or more computers or other devices. Generally, program modules  
11 include routines, programs, objects, components, data structures, etc. that perform  
12 particular tasks or implement particular abstract data types. Typically, the  
13 functionality of the program modules may be combined or distributed as desired in  
14 various embodiments.

### 15 16 **Computer Readable Media**

17 An implementation of the exemplary watermarking may be stored on or  
18 transmitted across some form of computer readable media. Computer readable  
19 media can be any available media that can be accessed by a computer. By way of  
20 example, and not limitation, computer readable media may comprise computer  
21 storage media and communications media.

22 Computer storage media include volatile and non-volatile, removable and  
23 non-removable media implemented in any method or technology for storage of  
24 information such as computer readable instructions, data structures, program  
25

1 modules, or other data. Computer storage media includes, but is not limited to,  
2 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,  
3 digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic  
4 tape, magnetic disk storage or other magnetic storage devices, or any other  
5 medium which can be used to store the desired information and which can be  
6 accessed by a computer.

7 Communication media typically embodies computer readable instructions,  
8 data structures, program modules, or other data in a modulated data signal such as  
9 carrier wave or other transport mechanism and included any information delivery  
10 media. The term "modulated data signal" means a signal that has one or more of  
11 its characteristics set or changed in such a manner as to encode information in the  
12 signal. By way of example, and not limitation, communication media includes  
13 wired media such as a wired network or direct-wired connection, and wireless  
14 media such as acoustic, RF, infrared, and other wireless media. Combinations of  
15 any of the above are also included within the scope of computer readable media.  
16

## 17 **Conclusion**

18 Although the improved stealthy audio watermarking has been described in  
19 language specific to structural features and/or methodological steps, it is to be  
20 understood that the improved stealthy audio watermarking defined in the appended  
21 claims is not necessarily limited to the specific features or steps described. Rather,  
22 the specific features and steps are disclosed as preferred forms of implementing  
23 the claimed improved stealthy audio watermarking.  
24  
25